Claims

What is claimed is:

1. A method comprising:

detecting an event associated with a device within a data processing system;

initiating a platform-independent device removal sequence for said device in response to detecting said event associated with said device;

virtually ejecting said device from said data processing system in response to initiating said platform-independent device removal sequence for said device; and

servicing said event associated with said device in response to virtually ejecting said device from said data processing system.

2. The method of claim 1, wherein initiating a platform-independent device removal sequence for said device in response to detecting said event associated with said device comprises generating a system control interrupt.

3. The method of claim 2, wherein generating a system control interrupt comprises generating a system control interrupt utilizing an INT_OUT command.

4. The method of claim 1, wherein said device comprises a processor including a first processor core and a second processor core, said method further comprising operating said first processor core and said second processor core in a functional redundancy check mode.

5. The method of claim 4, wherein detecting an event associated with a device within a data processing system comprises detecting an error within at least one of said first processor core and said second processor core.

6. The method of claim 4, said method further comprising disabling interrupts to said processor in response to initiating a platform-independent device removal sequence for said device.

7. The method of claim 4, wherein operating said first processor core and said second processor core in a functional redundancy check mode comprises concurrently executing a thread on said first processor core and said second processor core, said method further comprising:

saving a context of said thread in response to initiating a platform-independent device removal sequence for said device.

8. The method of claim 5, said method further comprising disabling said functional redundancy check mode in response to detecting said error within at least one of said first processor core and said second processor core.

9. The method of claim 8, wherein servicing said event associated with said device in response to virtually ejecting said device from said data processing system comprises:

resetting said processor; and

re-enabling said functional redundancy check mode in response to resetting said processor.

10. The method of claim 1, said method further comprising:

initiating a platform-independent device insertion sequence for said device in response to servicing said event associated with said device; and

virtually inserting said device into said data processing system in response to initiating said platform-independent device insertion sequence for said device.

11. A machine-accessible medium having machine-executable instructions embodied therein which, when executed by a machine, causes said machine to perform a method comprising:

detecting an event associated with a device within a data processing system;

initiating a platform-independent device removal sequence for said device in response to detecting said event associated with said device;

virtually ejecting said device from said data processing system in response to initiating said platform-independent device removal sequence for said device; and

servicing said event associated with said device in response to virtually ejecting said device from said data processing system.

12. The machine-accessible medium of claim 11, wherein said device comprises a processor including a first processor core and a second processor core, said method further comprising operating said first processor core and said second processor core in a functional redundancy check mode.

13. The machine-accessible medium of claim 12, wherein detecting an event associated with a device within a data processing system comprises detecting an error within at least one of said first processor core and said second processor core.

14. The machine-accessible medium of claim 12, said method further comprising disabling interrupts to said processor in response to initiating a platform-independent device removal sequence for said device.

15. The machine-accessible medium of claim 12, wherein operating said first processor core and said second processor core in a functional redundancy check mode comprises concurrently executing a thread on said first processor core and said second processor core, said method further comprising:

saving a context of said thread in response to initiating a platform-independent device removal sequence for said device.

16. The machine-accessible medium of claim 13, said method further comprising disabling said functional redundancy check mode in response to detecting said error within said at least one of said first processor core and said second processor core.

17. The machine-accessible medium of claim 16, wherein servicing said event associated with said device in response to virtually ejecting said device from said data processing system comprises:

resetting said processor; and

re-enabling said functional redundancy check mode in response to resetting said processor.

18. A data processing system comprising:

a processor; and

a memory coupled to said processor, said memory including a firmware interface to initiate a platform-independent device removal sequence for said processor in response to a detection of an event associated with said processor, to virtually eject said processor from said data processing system in response to an initiation of a platform-independent device removal sequence for said processor, and to service said event associated with said processor in response to a virtual ejection of said processor from said data processing system.

19. The data processing system of claim 18, wherein said firmware interface to initiate a platform-independent device removal sequence for said processor in response to a detection of an event associated with said processor comprises a firmware interface to generate a system control interrupt.

20. The data processing system of claim 19, wherein said firmware interface to generate a system control interrupt comprise a firmware interface to generate a system control interrupt utilizing an INT_OUT command.

21. The data processing system of claim 18, wherein said processor comprises a first processor core and a second processor core, said data processing system further comprising functional redundancy check logic to operate said first processor core and said second processor core in a functional redundancy check mode.

22. The data processing system of claim 21, wherein said firmware interface to initiate a platform-independent device removal sequence for said processor in response to a detection of an event associated with said processor comprises a firmware interface to initiate a platform-independent device removal sequence for

said processor in response to a detection of an error within at least one of said first processor core and said second processor core.

23. The data processing system of claim 21, said memory further including an operating system to disable interrupts to said processor in response to an initiation of a platform-independent device removal sequence for said processor.

24. The data processing system of claim 21, wherein

said functional redundancy check logic to operate said first processor core and said second processor core in a functional redundancy check mode comprises functional redundancy check logic to concurrently execute a thread on said first processor core and said second processor core; and

said memory further includes an operating system to save a context of said thread in response to an initiation of a platform-independent device removal sequence for said processor.

25. The data processing system of claim 21, wherein said firmware interface further comprises a firmware interface to disable said functional redundancy check mode in response to said detection of an event associated with said processor.

26. The data processing system of claim 25, wherein said firmware interface to service said event associated with said processor in response to a virtual ejection of said processor from said data processing system comprises a firmware interface to reset said processor and to re-enable said functional redundancy check mode in response to a reset of said processor.

27. The data processing system of claim 18, wherein said firmware interface further comprises a firmware interface to initiate a platform-independent device insertion sequence for said processor in response to a servicing of said event associated with said processor and to virtually insert said processor into said data processing system in response to an initiation of a platform-independent device insertion sequence for said processor.

28. An apparatus comprising:

a first firmware interface to detect an event associated with a processor within a data processing system;

a second firmware interface to initiate a platform-independent device removal sequence for said processor in response to a detection of said event associated with said processor;

a third firmware interface to virtually eject said processor from said data processing system in response to an initiation of a platform-independent device removal sequence for said processor; and

a fourth firmware interface to service said event associated with said processor in response to a virtual ejection of said processor from said data processing system.

29. The apparatus of claim 28, wherein said second firmware interface to initiate a platform-independent device removal sequence for said processor in response to a detection of said event associated with said processor comprises a firmware interface to generate a system control interrupt.

30. The apparatus of claim 28, said apparatus further comprising:

a fifth firmware interface to initiate a platform-independent device insertion sequence for said processor in response to a servicing of said event associated with said processor; and

a sixth firmware interface to virtually insert said processor into said data processing system in response to an initiation of a platform-independent device insertion sequence for said processor.